APPLICATION SECURITY

APP SEC

Application Security (AppSec) is full stack software security at every step in the Software Development and Data Lifecycles.
The subject is much broader than penetration testing and remediating vulnerabilities.

Training

AppSec requires good communication and the ability to impart knowledge to the wider engineering team.
Awareness, training and education of software engineers on security principles, secure architecture and secure coding.
To help them understand and identify threats, as well as guiding them in choosing appropriate security controls (encryption, authentication, authorization, etc.) to mitigate those threats.

Supply Chain Management

Before starting development, 3rd party component scanning will protect development workstations and production environments from malware.
To protect against any subsequent known vulnerabilities, scanning must continue throughout the development, deployment and operational lifecycle.

Design
(Security Requirements Analysis, Secure Architecture Design, Threat Modeling)

AppSec begins with the security requirements analysis of a system, we solicit these through use cases, abuse cases and security use cases; or through user stories, abuse stories and security user stories.

The Application Security Verification Standard (ASVS) can be a useful guideline.
We must capture the requirements, define test cases, and track them through to release in a Security Requirements Traceability Matrix (SRTM).

AppSec is building security into the design, reviewing and Threat Modeling the architecture iteratively, ensuring the design implements sound security principles, controls, and mitigates any threat.

Implementation
(Secure Coding, SAST, SCA, Security Review of both Requirements and Code)

AppSec is automation of security quality gates to check the code is being written securely.

Static Application Security Testing (SAST) checks for vulnerabilities in the code. Software Composition Analysis (SCA) checks the 3rd party software component versions being used to ensure they don't contain known vulnerabilities.

AppSec is reviewing security before QA testing, checking that requirements are being verified by the tests being carried out, the right controls are in place and those controls are working as expected.

There should be a manual code review of parts of the system that are security sensitive and white-box testing of any areas with potential flaws.

Testing
(Security Testing, IAST, DAST, Penetration Testing)

AppSec is in QA, automating Security Testing Dynamic Application Security Testing (DAST) and Interactive Application Security Testing (IAST) are fundamental.

IAST and DAST involve testing the running application. IAST instruments the running application so that the engineer can track the behaviour of the application while security testing.

Both are often complex and problematic when dealing with Single Page applications or applications that make use of dependency injection.

Operational Security

24h

AppSec is ensuring the implementation of Secure Deployment principles.
A change management and configuration management process should be in place.
Ensuring credentials are the right ones for each environment, protecting against risks such as debug logging being active or exceptions being visible to users in production.

Runtime Application Self Protection (RASP) can protect against Zero Day vulnerabilities by allowing you to hot patch the issues in production while a remediation is being developed.

Support AppSec Support Engineering

HELP

AppSec requires partnership with Engineering. With an industry average of 1 AppSec engineer to 100 software engineers there just isn't the bandwidth to do everything without effective collaboration.

OWASP

Brett Crawley